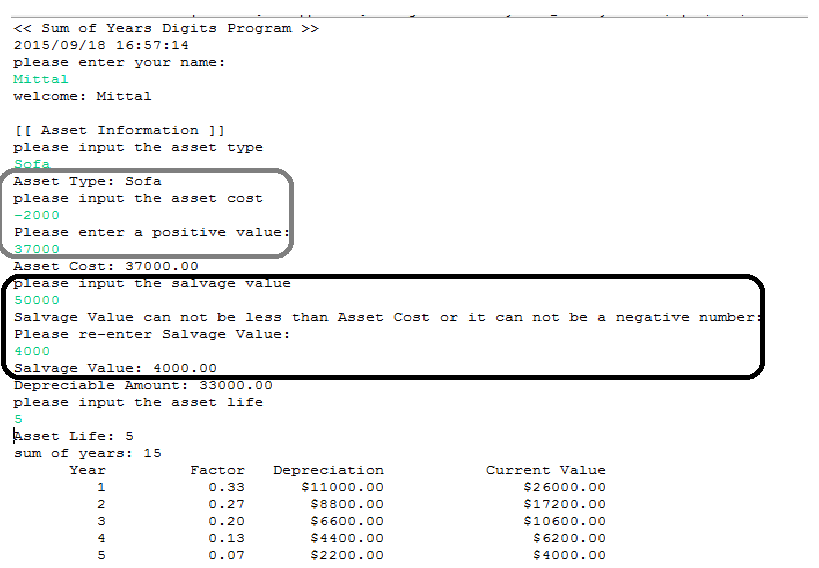
**PROJECT Control Structures in Java - Computing Depreciation**

# **First Example: Positive Testing**

# 

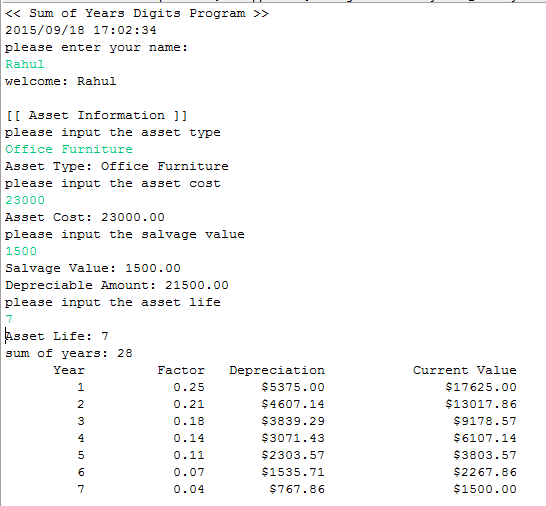
# **Scenario I : Including negative testing with Exception Handling**

|  |  |
| --- | --- |
| ***asset type*** | **Sofa** |
| ***asset cost*** | $ 37,000.00 |
| ***salvage value*** | $ 4,000.00 |
| ***asset life*** | 5 years |
|  |  |



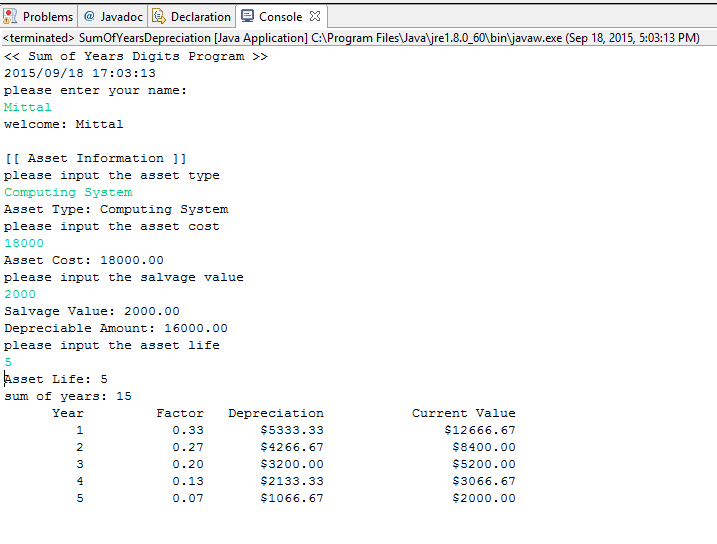
# **Scenario 2:**

|  |  |
| --- | --- |
| ***asset type*** | **Office Furniture** |
| ***asset cost*** | $ 23,000.00 |
| ***salvage value*** | $ 1,500.00 |
| ***asset life*** | 7 years |
|  |  |

ree

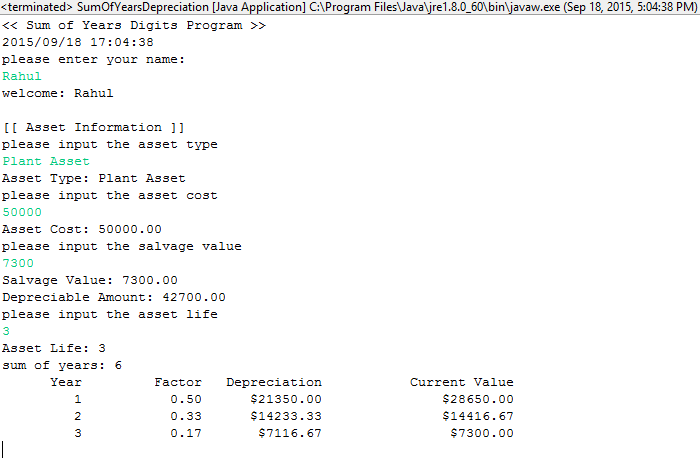
# **Scenario III:**

|  |  |
| --- | --- |
| ***asset type*** | **Computing System** |
| ***asset cost*** | $ 18,000.00 |
| ***salvage value*** | $ 2,000.00 |
| ***asset life*** | 5 years |
|  |  |



# **Scenario IV**

|  |  |
| --- | --- |
| ***asset type*** | **Plant Asset** |
| ***asset cost*** | $ 50,000.00 |
| ***salvage value*** | $ 7,300.00 |
| ***asset life*** | 3 years |
|  |  |



**Source Code:**

\*\*

\* Name: Rahul Mittal

\* Lab Name: ITMD\_510\_LAB\_02

\* Current Date: 09/18/2015

\*/

**import** java.text.DateFormat;

**import** java.text.DecimalFormat;

**import** java.text.SimpleDateFormat;

**import** java.util.Date;

**import** java.util.Scanner;

**public** **class** SumOfYearsDepreciation

{

// the global variables are declared

**static** **double** *assetCost* = 0;

**static** **double** *salvageValue* = 0;

**static** **double** *depreciableAmount* = 0;

**static** **int** *assetLife* = 0;

// declare a Scanner class object

**static** Scanner *sc* = **new** Scanner(System.***in***);

// declare a DecimalFormat class object

**static** DecimalFormat *two* = **new** DecimalFormat("0.00");

/\*\*

\* Validate the values of Asset Cost & Salvage Value.

\* Salvage value can not be greater than Asset Cost at any given point

\* If the Asset Cost is less than Salvage Value, throw an error

\* **@param** assetCost Depreciable Asset Cost

\* **@param** salvageValue Salvage Value entered in the beginning

\*/

**public** **static** **void** CheckDepreciation(**double** assetCost, **double** salvageValue)

{

**if**( assetCost < salvageValue)

{

System.***out***.println("AssetCost is less than Salvage Value" + "\t" + assetCost + "\t" + salvageValue);

System.*exit*(0);

}

}

/\*\*

\* Get all the inputs from the user and validate it properly

\* Asset Cost can not be in negative

\* Salvage Value can not exceed Asset Cost and can not be in negative

\*/

// method to receive asset information

**public** **static** **void** AssetInfo()

{

// declare and initialize a variable

String assetType = "";

// display output block information

System.***out***.println("[[ Asset Information ]]");

// request, receive and echo the asset type

System.***out***.println("please input the asset type");

assetType = *sc*.nextLine();

System.***out***.println("Asset Type: " + assetType);

// request, receive, echo the asset cost, salvage value

System.***out***.println("please input the asset cost");

**try**

{

*assetCost* = *sc*.nextDouble();

}

//Validate Asset Cost and if it's negative ask user to re-enter

**finally**

{

**while** (*assetCost*<= 0) {

System.***out***.println("Please enter a positive value: ");

*assetCost* = *sc*.nextDouble();

}

}

System.***out***.println("Asset Cost: " +

*two*.format(*assetCost*));

System.***out***.println("please input the salvage value");

**try**{

*salvageValue* = *sc*.nextDouble();

}

//Validate Salvage Value, it can neither be more than asset cost nor can be negative

**finally**

{

**while** (*salvageValue* <= 0 || *salvageValue* > *assetCost*) {

System.***out***.println("Salvage Value can not be less than Asset Cost or it can not be a negative number: ");

System.***out***.println("Please re-enter Salvage Value: ");

*salvageValue* = *sc*.nextDouble();

}

}

System.***out***.println("Salvage Value: " +

*two*.format(*salvageValue*));

// compute, echo depreciable amount as (cost - salvage)

*depreciableAmount* = *assetCost* - *salvageValue*;

System.***out***.println("Depreciable Amount: " +

*two*.format(*depreciableAmount*));

// request, receive and echo the asset life

System.***out***.println("please input the asset life");

**try** {

*assetLife* = *sc*.nextInt();

}

**finally**

{

**while** (*assetLife*<= 0) {

System.***out***.println("Please enter a positive value: ");

*assetLife* = *sc*.nextInt(); }

}

System.***out***.println("Asset Life: " + *assetLife*);

}

// method to sum the years

**public** **static** **int** GaussSum(**int** num)

{

// declare and initialize a variable

**int** sumOfYears = 0;

// use Gauss Formula to sum the years

sumOfYears = num \* (num + 1) / 2;

// echo the sum of years

System.***out***.println("sum of years: " + sumOfYears);

// return the sum

**return** sumOfYears;

}

/\*\*

\* First Calculate the Factor for every year

\* Calculate the depreciable amount for every year

\* Print all the values

\* **@param** assetCost - Amount getting reduced after every iteration

\* **@param** depreciableAmount - amount to be deducted from asset cost

\* **@param** assetLife - Number of years of asset life

\* **@param** sumOfYears - Sum of years to calculate factor

\*/

**public** **static** **void** ShowDepreciationSchedule

(**double** assetCost, **double** depreciableAmount, **int** assetLife, **int** sumOfYears)

{

**int** temp = 0;

System.***out***.printf("%10s%15s%15s%25s", "Year", "Factor" , "Depreciation", "Current Value" + "\n");

//start iteration for every year deducting 1 every year

**for**( **float** i=assetLife; i>=1; i--)

{

temp++;

**double** factor = i/sumOfYears;

**double** depreciable = factor \* depreciableAmount;

assetCost = assetCost - depreciable;

//convert all the values in string to bring a proper format in output

String s\_Fact = String.*format*("%.2f", factor);

String s\_Depreciable = String.*format*("%.2f", depreciable);

String s\_assetCost = String.*format*("%.2f", assetCost);

DecimalFormat df = **new** DecimalFormat("#.##");

**double** temp\_sal\_val = Double.*valueOf*(df.format(*salvageValue*));

**double** temp\_asset = Double.*valueOf*(df.format(assetCost));

//Check the value of Asset Cost and Salvage value every time

//if the value of asset cost is less than salvage value

//throw an error and exit

*CheckDepreciation* (temp\_asset,temp\_sal\_val);

System.***out***.format ("%10d%15s%15s%25s", temp, s\_Fact, "$"+s\_Depreciable, "$"+s\_assetCost + "\n");

}

}

**public** **static** **void** main(String[] args)

{

// declare and initialize the local variable(s)

String userName = "";

// display output block information

System.***out***.println("<< Sum of Years Digits Program >>");

//Print Today's date

DateFormat dateFormat = **new** SimpleDateFormat("yyyy/MM/dd HH:mm:ss");

Date date = **new** Date();

System.***out***.println(dateFormat.format(date)); //2015/09/15 14:19:25

// meet and greet the program user

System.***out***.println("please enter your name: ");

userName = *sc*.nextLine();

System.***out***.println("welcome: " + userName + "\n");

// call the AssetInfo() method

*AssetInfo*();

// call the GaussSum() method

**int** sumOfYears = *GaussSum*(*assetLife*);

// call the ShowDepreciationSchedule() method

*ShowDepreciationSchedule*(*assetCost*, *depreciableAmount*, *assetLife*, sumOfYears);

}

}